

CLAIMS:

What is claimed is:

Claim 1:

1 1. In a computer system having identifiable memory address spaces which may be
2 allocated to processes, a queue bank repository system for facilitating control of such
3 address spaces to such processes comprising:

4 a repository of said memory address spaces wherein each memory address
5 space within said repository may be called an entry and each said entry contains a
6 value which can be either an end-of-repository space indicator or a current value
7 which indicates a next available entry, and wherein said repository has a header
8 memory value which is either a one of said current values or said end-of-
9 repository space indicator.

10 a manager of said repository comprising a functionality for generating a
11 token wherein said token contains an indication of said header memory value and
12 for passing said token to a client process if said client process passes or attempts
13 to pass a queue bank to said manager for storage into one of said memory
14 address spaces of said repository, and wherein said manager further comprises a
15 functionality for storing said queue bank reference from said client process into a
16 one of said memory address spaces indicated by said header memory value.

Claim 2:

1 2. The queue bank repository system of claim 1 wherein said functionality for
2 generating a token is an instruction and wherein said functionality for storing said queue
3 bank reference from said client process into a one of said memory address spaces
4 indicated by said header memory value is also an instruction.

Claim 3:

1 3. The queue bank repository system of claim 1 wherein said manager further
2 comprises a functionality for reading a value in said one of said memory address spaces
3 indicated by said header value and storing said value into said header prior to or
4 contemporaneously with storing said queue bank reference from said client process into

5 said one of said memory address spaces indicated by said current value so that following
6 said storing, said current value shall contain a value that was in said one of said memory
7 address spaces.

Claim 4:

1 4. The queue bank repository system of claim 1 wherein said functionality for reading
2 further comprises a functionality for clearing said value from said one of said memory
3 address spaces while storing said value into said header current value.

Claim 5:

1 5. The queue bank repository system of claim 1 wherein said manager further
2 comprises a functionality for reading a value in said one of said memory address spaces
3 indicated by said header value and if said header value is said end-of-repository value,
4 comprising a functionality for generating a status indicating the repository is full.

Claim 6:

1 6. The queue bank repository system of claim 1 wherein said manager further
2 comprises a functionality for reading a value in said one of said memory address spaces
3 indicated by said header value and if said header value is said end-of-repository value,
4 comprising a functionality for calling an operating system to cause said operating system
5 to allocate a new space to add to said repository and for returning a value indicative of
6 the address of the entry in said new space wherein queue bank references can be stored
7 by a client process.

Claim 7:

1 7. The queue bank repository system of claim 1 wherein said repository of said
2 memory address spaces has more than one set of said memory address spaces and uses
3 only a portion of those of said more than one sets as said memory address space within
4 said repository unless said portion becomes fully occupied and in such event, upon said
5 client process attempting to pass or passing a queue bank reference to said manager for
6 storage to said portion when said portion is fully occupied, a functionality extends said

7 memory address space within said repository to include another one of said one set of
8 said memory address spaces to said memory address space.

Claim 8:

1 8. The queue bank repository system of claim 1 wherein a said token is available in a
2 plurality of different formats as determined by the specific implementation. Said formats
3 include, but are not limited to, pointer, offset, and entry index.

Claim 9:

1 9. The queue bank repository system of claim 1 wherein said tokens within the
2 repository each have a size to hold any queue bank reference that could be generated on
3 the system.

Claim 10:

1 10. A method of storing a queue bank reference from a client process into a queue
2 bank repository comprising:

3 indicating that a client process needs to store a queue bank reference into
4 said queue bank repository,

5 providing to said client process a token having an indication of an entry
6 address into which the queue bank reference is stored in said queue bank
7 repository such that the client can later retrieve the stored queue bank,

8 storing said queue bank reference into said entry address, and

9 removing said queue bank from the visible address space of the client
10 process.

Claim 11:

1 11. The method of claim 10 further comprising,

2 reading from a header in said queue bank repository a next available entry
3 address location prior to providing said token to said client and wherein said
4 storing step comprises storing said queue bank reference into a last available
5 entry address location.

Claim 12:

1 12. The method of claim 10 further comprising manufacturing said token to include an
2 indication of said last available entry address location into which said client queue bank
3 reference was stored.

Claim 13:

1 13. The method of claim 10 further comprising manufacturing said token to include an
2 indication of said last available entry address location into which said client queue bank
3 reference was stored, or if the repository is full, providing an indication of fullness.

Claim 14:

1 14. The method of claim 10 further comprising manufacturing said token to include an
2 indication of said last available entry address location into which said client queue bank
3 reference was stored, or if the repository is full, not providing any token until said
4 repository has an available address entry.

Claim 15:

1 15. The method of claim 10 further comprising manufacturing said token to include an
2 indication of said last available entry address location into which said client queue bank
3 reference was stored, or if the repository is full, providing an interrupt to an operating
4 system.

Claim 16:

1 16. The method of claim 15 wherein said operating system provides for more available
2 entry address locations when it receives said interrupt.

Claim 17:

1 17. The method of claim 10 further comprising manufacturing said token to include an
2 indication of said last available entry address location into which said client queue bank
3 reference was stored, or if the repository is full, opening a new space of entries via a call
4 to an operating system, so that said manufacturing of said token can be accomplished
5 with an indication that said client queue bank reference was stored in said new space.

Claim 18:

1 18. A method of retrieving a queue bank by a client process from a queue bank
2 repository comprising:
3 providing a token to said queue bank repository by said client,
4 reading said token to determine an address containing said queue bank
5 reference by said queue bank repository,
6 providing data from said address containing said queue bank reference to
7 said client process by said queue bank repository, and
8 establishing said retrieved queue bank in the visible address space of the
9 client as specified by the client.

Claim 19:

1 19. A method for handling invalid attempts to retrieve a queue bank by a client
2 process from a Queue Bank Repository comprising:
3 providing a false token to said queue bank repository by said client,
4 reading said token to determine an address containing said queue bank
5 reference by said queue bank repository
6 providing a status indicating that the token was not valid (no deposit
7 currently exists at that token address).

Claim 20:

1 20. A system for handling a queue bank repository system comprising at least two
2 methods, the first method, for storing a queue bank from a client process into a queue
3 bank repository comprising:
4 indicating that a client process needs to store a queue bank into said
5 queue bank repository,
6 providing to said client process a token having an indication of an entry
7 address into which the queue bank reference is stored in said queue bank
8 repository such that the client can later retrieve the stored queue bank, and
9 storing said queue bank reference into said entry address,
10 and

11 the second method, for retrieving a said queue bank that has been from a client
12 process in a queue bank repository comprising:
13 providing a token to said queue bank repository by said client,
14 reading said token to determine an address containing said queue bank
15 reference by said queue bank repository, and
16 providing data from said address containing said queue bank to said client
17 process by said queue bank repository.

Claim 21:

1 21. The system of claim 19, further comprising a method for handling invalid attempts
2 to retrieve a queue bank by a client process from a Queue Bank Repository comprising:
3 providing a false token to said queue bank repository by said client,
4 reading said token to determine an address containing said queue bank by
5 said queue bank repository, and
6 providing a status indicating that the token was not valid.